

# EXHIBIT C

[MAIN MENU](#) [MY STORIES: 25](#) [FORUMS](#) [SUBSCRIBE](#) [JOBS](#)

## GEAR & GADGETS / PRODUCT NEWS & REVIEWS

### The Play Store comes to Chrome OS, but not the way we were expecting

Under the hood, Google takes a brand new approach to Android apps on Chrome OS.

by **Ron Amadeo** - May 19, 2016 11:00am PDT

115



The Android game *Galaxy On Fire* running on the Chromebook Pixel.  
Google

MOUNTAIN VIEW, Calif.—It's really happening. Android apps are coming to Chrome OS. And it's not just a small subset of apps; the entire Google Play Store is coming to Chrome OS. More than 1.5 million apps will come to a platform that before today was "just a browser," and Android and Chrome OS take yet another step closer together.

In advance of the show, we were able to sit down with members of the Chrome OS team and get a better idea of exactly what Chrome OS users are in for. The goal is an "It just works" solution, with zero effort from developers required to get their Android app up and running. Notifications and in-line replies should all work. Android apps live in native Chrome OS windows, making them look like part of the OS. Chrome OS has picked up some Android tricks too—sharing and intent systems should work fine, even from one type of app or website to another. Google is aiming for a unified, seamless user experience.

Starting in early June, developer channel builds of Chrome OS will see a pop-up message allowing them to opt-in to Google Play and Android app compatibility. This will roll out to touch-enabled Chromebooks first in the "M53 Dev" version, with support for non-touch devices coming soon after. We were told a full-scale rollout to the Chrome OS stable channel should happen sometime in September or October.

Of course, not everyone will be welcoming of Android apps on Chrome OS. Schools and businesses in particular value Chrome OS specifically because it does not have traditional apps—with no apps, there is nothing evil for users to install that will mess something up. Google is trying to keep these users happy, too, with lots of control over the feature for administrators. Google is promising Android app

#### LATEST FEATURE STORY [▲](#)



[FEATURE STORY \(2 PAGES\)](#)

#### Racing with royalty —Behind the scenes with Pikes Peak's "King of the Mountain"

We embed with the team as it aims to be fastest overall at the event's 100th anniversary.

#### WATCH ARS VIDEO [▲](#)

#### HP Spectre 13 reviewed: Remarkably thin, but not light on power

Trying to out-svelte the svelte masters, HP's shiny notebook starts at \$1,169.

#### STAY IN THE KNOW WITH [▲](#)

#### LATEST NEWS [▲](#)

[FIXER UPPER](#)

**Volkswagen lawyer says 3.0L diesel cars will likely be fixed, not bought back**

[DEJA VU](#)

**Dell gets out of the Android business, and everything old is new again**

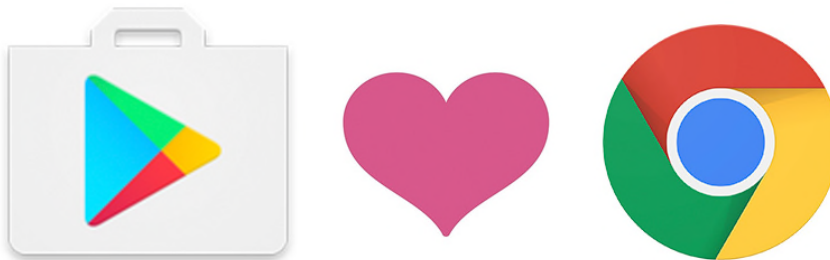


**Report: Apple blocks Spotify app update after in-app subscription removal**

controls with "multiple levels of granularity" for Chrome OS administrators. For companies, administrators will need to explicitly enable the feature in order for their users to access Android apps, so organizations that want to keep their Chromebooks simple can still do so. If admins opt to enable Android apps, they can throw open the floodgates to the whole Play Store or enforce an apps whitelist.

Chrome OS now will have two app stores: Google Play and the Chrome Web Store, but the Chrome team said nothing was going to happen to the Chrome Web Store—it will continue to house Chrome themes and extensions.

## Goodbye ARC; Hello containers



We can't say that the announcement of Google Play on Chrome OS is a surprise. Google first started publicly experimenting with Android apps on Chrome OS in 2014 with ARC, the [App Runtime for Chrome](#). ARC implemented the Android runtime on top of Chrome's "Native Client" extension architecture, and Google worked with some developers to port a handful of Android apps to Chrome OS. Given that the Native Client extension system was designed with portability in mind, ARC ran on every platform that supports Chrome extensions, so this small amount of Android apps could run on Chrome OS, Windows, Mac, and Linux.

The real shocker here is that this release of Google Play on Chrome OS is *not* based on ARC. Zelidrag Hornung, the engineering director of Chrome & Android, filled us in on the details: "We have redone this completely differently. There are no connecting points between the two projects (ARC and today's announcement) from an implementation perspective." ARC wasn't good enough, so Google started over from scratch. "With the initial version of ARC, we managed to push things forward, but fundamentally developers still needed to do serious work to their application to make it work in that environment," Hornung told Ars. "With this new model, there is virtually no work that an Android Developer needs to do with their app. They just publish it to the Play Store and it just works."

ARC couldn't pass Google's own Android Compatibility Test Suite (CTS), and Hornung says it would have been "very hard to make it CTS compliant." Apps written with the NDK—an alternative to the Android SDK that allows developers to use languages other than Java—wouldn't work with ARC, which locked most games out of the project. "It was pretty clear that we had to take a radical step forward and make something that was just Android from a developer perspective," Hornung said. "That's why we started to implement this from scratch."

The new model dumps the native-client based implementation for an unmodified copy of the Android Framework running in a container. Containers usually bundle an app up with all of its dependencies, like the runtime, libraries, binaries, and anything else the app needs to run. This allows the difference between application environments to be abstracted away. In this case, Google is putting the entire Android Framework into a container, all the way down to the Hardware Abstraction Layer.

It's a lot like virtualization, but a virtual machine would sandbox the apps away from other apps and the rest of the OS. Containers let the Android apps access the underlying OS, which allows them to communicate with each other. Connectivity, storage, audio, touch, and all the other inputs and outputs get connected to Chrome OS, allowing Android to interact with the outside world. Containerization is also designed to be lightweight—it doesn't use anywhere near the storage resources virtualization uses, since you aren't including a copy of the OS with every single app. The apps can all share a single Android app framework. Containers also don't need to allocate individual user storage for each instance.

:(

**Tesla's Autopilot being investigated by the government following fatal crash**

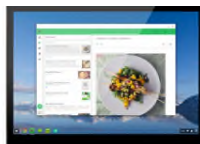
THAT AXE IS HUGE

**Rare Viking "death house" discovered in Denmark**

WHEN ALL ELSE FAILS...

**People eat healthier when real-life emojis literally point them to produce**

### FURTHER READING



#### GOOGLE'S ARC BETA RUNS ANDROID APPS ON CHROME OS, WINDOWS, MAC, AND LINUX

"App Runtime for Chrome" takes a big step toward making Android a universal binary.

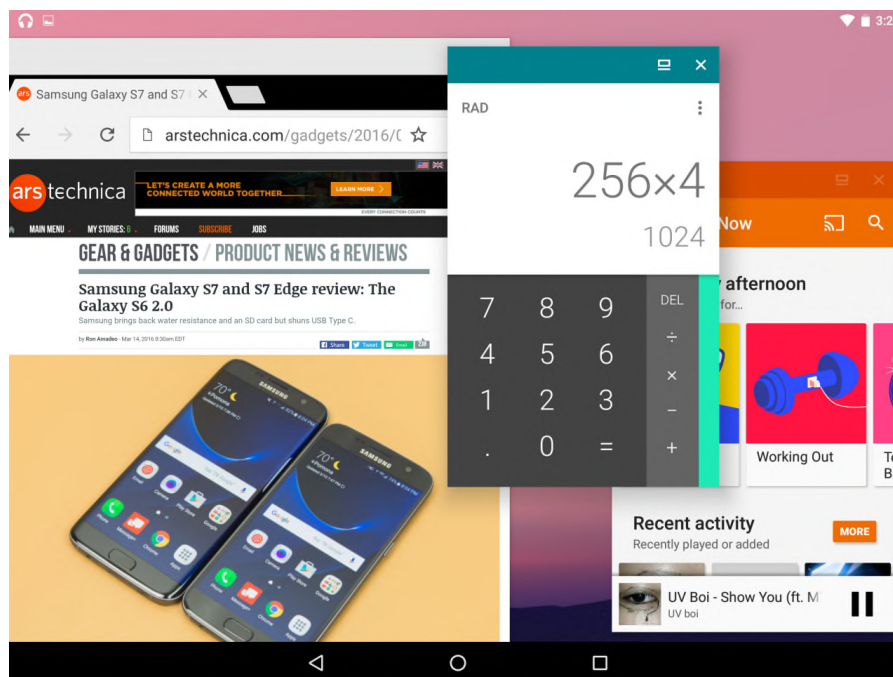
The Chrome OS Android app framework gets updated directly by Google, just like a Nexus device, and the Chrome OS team has committed to keeping pace with Android's monthly security update program. Thanks to the full control over Chrome OS and centralized update system, the Chrome team expects nearly 100 percent of Android-enabled Chromebooks to quickly update to the newest version of Android. (If only Android phones could come close to that.)

Native apps written with the NDK should work just fine, since, just like Android phones, Chromebooks run on a mix of x86 and ARM. x86 isn't as prevalent on the Android side as it is on Chromebooks, but newer versions of the Android NDK have been automatically spitting out Intel binaries alongside ARM for some time now. "In reality, we expect about one quarter of the apps to not have Intel [x86] binaries," Hornung told Ars, but the Android Framework has long shipped with binary translation technology to ensure apps are covered when Intel binaries are not present. There is some speed loss when translating ARM to x86, but it shouldn't be a problem given that most x86 Chromebooks are going to be much faster than an ARM phone.

The Chrome OS team says it "leveraged lots of work from the open source community" to make this possible, but for now, there aren't any plans to open source the work. "The Android side here is not open source," Hornung explained; "currently, it allows us to move faster." Puneet Kumar, engineering director of Chrome, chimed in, "We honestly didn't open it because we weren't sure how well it was going to work."

Hornung continued, "I think this is an open question what we're going to do with it when we launch it, but at this point we don't have any plans to open source it immediately."

## Bring on the (Desktop) Android apps



[Enlarge](#) / Android N Developer Preview 2 running the hidden freeform window mode.

Ron Amadeo

Chrome OS will now be home to more than 1.5 million Android apps, most of which are designed for smartphones. Phone apps are a disaster on tablets, because they stretch the app to fill the screen. On desktops, though, when you can have everything in a floating window, a phone-sized phone app **isn't so bad**. The rare tablet app is a nice bonus, too.

For now, Google's test platform is running on Android 6.0 Marshmallow, because Android N just isn't done yet, but there are critical features in Android N that will help out this Chrome OS implementation. With **split screen** and the **hidden floating window functionality**, Android N is the first version of the OS that asks developers to deal with having their apps resized. Those app improvements will come to Chrome OS, too, so we should see updated apps seamlessly adjust from phone to tablet mode when resized.

It's not just about phone and tablet apps, though. Chrome OS is now the first platform for desktop Android apps, and it's something the team is focusing on getting developers involved in. "We're putting a lot of that [desktop Android] documentation together right now," Chrome's

FURTHER READING

product manager, Kan Liu, told Ars. "One of the reasons we're announcing at I/O is because we want to get it out a little bit early, so developers can go and try their apps on this and make sure everything works smoothly. Then they can optimize things for multiwindow, for keyboard and trackpad and so forth, so when we roll this out to stable channel and more end users later this year they can be ready."

"We also added some of the productivity framework features for Android developers," Hornung added. "We defined how a resizable application should behave so Android developers in N will be able to make those applications more fluid in a desktop environment. And also we've done some cross application things like drag-and-drop for complex content. There are things that are coming in N that will allow developers to improve this experience and do richer things in the desktop." Other things, like pinch zoom and two-finger scrolling, have already been mapped to the Chrome OS trackpad.

So that's it. Android Desktop apps are a real thing now, just locked in Chrome OS. Will we ever see an Android OS that runs these desktop Android apps on a big screen with a mouse and keyboard? Android apps were once referred to as a "universal application platform," which makes it sound like Google has big plans for its little phone operating system.



#### ANDROID ON THE DESKTOP: NOT REALLY "GOOD," BUT BETTER THAN YOU'D THINK

Surprisingly good mouse and keyboard support, but the interface can't scale.

READER COMMENTS 115



**Ron Amadeo** / Ron is the Reviews Editor at Ars Technica, where he specializes in Android OS and Google products. He is always on the hunt for a new gadget and loves to rip things apart to see how they work.

[@RonAmadeo on Twitter](#)

← OLDER STORY

NEWER STORY →

YOU MAY ALSO LIKE ↴

#### SITE LINKS

[About Us](#)  
[Advertise with us](#)  
[Contact Us](#)

#### MORE READING

[RSS Feeds](#)  
[Newsletters](#)

#### CONDE NAST SITES

[Reddit](#)  
[Wired](#)  
[Vanity Fair](#)

[Reprints](#)

[Visit Ars Technica UK](#)

[Style](#)

[Details](#)

## SUBSCRIPTIONS

[Subscribe to Ars](#)

[Visit our sister sites](#)

[Subscribe to a magazine](#)

[VIEW MOBILE SITE](#)

CONDÉ NAST

WIRED Media: Ars Technica and WIRED

© 2016 Condé Nast. All rights reserved

Use of this Site constitutes acceptance of our [User Agreement](#) (effective 1/2/14) and [Privacy Policy](#) (effective 1/2/14), and [Ars Technica Addendum](#) (effective 5/17/2012)

[Your California Privacy Rights](#)

The material on this site may not be reproduced, distributed, transmitted, cached or otherwise used, except with the prior written permission of Condé Nast.

[Ad Choices](#)